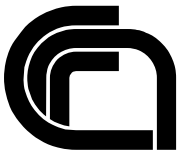

Consiglio Nazionale delle Ricerche

Performance analysis of maximum likelihood methods for regularization problems with nonnegativity constraints

P. Favati G. Lotti O. Menchi F. Romani

IIT TR-13/2010

Technical report

Febbraio 2010

Istituto di Informatica e Telematica

Performance analysis of maximum likelihood methods for regularization problems with nonnegativity constraints

P. Favati G. Lotti O. Menchi F. Romani

Abstract

In many numerical applications, for instance in image deconvolution, the nonnegativity of the computed solution is required. When a problem of deconvolution is formulated in a statistical frame, the recorded image is seen as the realization of a random process, where the nature of the noise is taken into account. This formulation leads to the maximization of a likelihood function which depends on the statistical property assumed for the noise. In this paper we revisit, under this unifying statistical approach, some iterative methods coupled with suitable strategies for enforcing nonnegativity and other ones which instead naturally embed nonnegativity. For all these methods we carry out a comparative study taking into account several performance indicators. The reconstruction efficiency, the computational cost, the consistency with the discrepancy principle (a common technique for guessing the best regularization parameter) and the sensitivity to this choice are compared in a simulated context, by means of an extensive experimentation on both 1D and 2D problems.

Keywords: Regularization Problems, Image Deconvolution, Maximum Likelihood Methods, Nonnegativity Constraints.

1 Introduction

In this paper we deal with the numerical solution of discrete ill-posed problems which arise from the discretization of a Fredholm integral equation of the first kind

$$g(s) = \int K(s,t) f(t)\, dt, \tag{1}$$

where the square integrable kernel $K(s,t)$ and the right-hand side $g(s)$ are given functions and $f(t)$ is the unknown solution. A typical 2D example of such a problem is the image deconvolution problem, where $f(t)$ and $g(s)$ represent a real object and its image respectively, and $K(s,t)$ represents

the imaging system and is responsible for the blurring of the image. In many applications the blurred image $g(s)$ is not available, being replaced by a finite set $\boldsymbol{g}$ of measured quantities, and is degraded by the noise which affects the process of image recording. For instance, in the deconvolution of astronomical and medical images a counting process is involved, of photons in the first case and of the rays emitted by body organs or by some injected substance in the second case. The noise is mainly due to the fluctuations in this counting process, which obeys to Poisson statistics. But there is also the readout noise, due to imperfections of the recording device, which obeys Gaussian statistics.

Discretized problems arising from equation (1) are frequent also in 1D contexts, for example in signal processing and in the computation of inverse transformations ([11] gives a collection of such problems).

The problem of restoring $f(t)$ from $\boldsymbol{g}$ is an ill-posed problem and the linear system which is obtained when equation (1) is discretized inherits the ill-posedness: the resulting matrix is highly ill-conditioned, and regularization methods must be used to solve the system [12]. Another important feature of the problem is the nonnegativity of the functions involved in (1) and we expect the solution of the linear system to be nonnegative. Enforcing such a constraint is not an easy task. Iterative methods (see [20] for a general presentation), often applied as regularization techniques, may give solutions with negative entries.

When the problem of deconvolution is formulated in a statistical frame, the data are seen as the realization of a random process, where the nature of the noise is taken into account. Of course the noise is in general not known, but the knowledge of some basic statistical property may be assumed. This formulation leads to the maximization of a likelihood function which depends on the assumed probability statistics. In this paper we revisit, under the unifying approach of the maximization of a likelihood function, some iterative methods coupled with suitable strategies for enforcing nonnegativity and other ones which instead naturally embed nonnegativity. Our aim is to compare the performances of these methods; many of them can be seen as belonging to the framework of Scaled Gradient Projection method (SGP).

Nearly all the papers that deal with deconvolution present results comparing the performances of different methods. Comparisons of the methods can be found for example in [1], [2],[3], [4], [5], [6], [7], [15], [16], [17], [19]. In general, the comparisons take into account for some selected problems the relative error of the best reconstructed solution and the cost required to obtain it. We feel that the subject deserves a more systematic investigation, taking into consideration also different aspects, as for instance the possibility to use a stopping rule based on the discrepancy principle, which is an important element of the success of a method. We carry out a comparative study of the methods taking into account several performance indicators. The computational cost, the reconstruction efficiency, the consistency with

the discrepancy principle (as standard technique for choosing the best regularization parameter) and the sensitivity to this choice are compared in a simulated context.

The problem is presented in Section 2, together with the maximum likelihood approach, the SGP framework is introduced in Section 3, the methods taken into consideration are listed and discussed in Section 4. Sections 5 and 6 are dedicated to the performance measures and experimental results.

2 The problem

Let

$$\boldsymbol{b}^* = A\boldsymbol{x}^* \tag{2}$$

be the discretized version of equation (1), with $\boldsymbol{b}^*, \boldsymbol{x}^* \in \mathbb{R}^N$ and $A \in \mathbb{R}^{N\times N}$. The matrix A is assumed to be severely ill-conditioned with singular values decaying to zero without significant gap to indicate numerical rank. Since vector $\boldsymbol{b}^*$ is not exactly known and only the noisy vector

$$\boldsymbol{b} = \boldsymbol{b}^* + \boldsymbol{\eta}$$

is available, finding a good approximation of $\boldsymbol{x}^*$ by means of the system

$$A\boldsymbol{x} = \boldsymbol{b} \tag{3}$$

is an ill-posed problem.

In the image deconvolution problems the N-vector $\boldsymbol{x}^*$ stores columnwise the pixels of an $n \times n$ object, with $N = n^2$, and $\boldsymbol{b}^*$ stores analogously the blurred image. Hence the ith component of the vectors $\boldsymbol{x}^*$, $\boldsymbol{b}^*$ and $\boldsymbol{b}$ represents respectively the light intensity or the radiation emitted by the ith pixel of the object, arriving at the ith pixel of the blurred image and recorded in the ith pixel of the noisy image. The component a_{ij} of matrix A measures the fraction of the light or of the rays emitted by the ith pixel of the object which arrives at the jth pixel of the image. Then all the quantities involved, A, $\boldsymbol{x}^*$, $\boldsymbol{b}^*$ and $\boldsymbol{b}$, are assumed componentwise nonnegative and (3) is replaced by the constrained problem

$$\begin{cases} A\boldsymbol{x} = \boldsymbol{b}, \\ \boldsymbol{x} \geq \boldsymbol{0}. \end{cases} \tag{4}$$

We assume that both $A\boldsymbol{x} \neq \boldsymbol{0}$ and $A^T\boldsymbol{x} \neq \boldsymbol{0}$ for any $\boldsymbol{x} \geq \boldsymbol{0}$ with $\boldsymbol{x} \neq \boldsymbol{0}$, and that $A\boldsymbol{e} > \boldsymbol{0}$ and $A^T\boldsymbol{e} > \boldsymbol{0}$, where $\boldsymbol{e}$ is the vector of all ones (i.e. the sums by rows and columns of A are all nonzero).

The computation of $A\boldsymbol{x}$ or $A^T\boldsymbol{x}$ provides the major part of the computational cost of the methods we will consider. In the 2D case matrix A has frequently a 2-level Toeplitz structure, which reduces to a 2-level circulant structure when periodic boundary conditions are set. In this way the

matrix-vector product can be easily computed by means of FFT. In the 1D case the size of the problem is generally smaller and the matrix-vector product can be performed directly. In the experiments we will consider examples of both kinds: large 2D problems with 2-level circulant matrices and smaller but severely ill-conditioned 1D problems.

Because of the presence of the noise, the solution $\boldsymbol{x}_\eta = A^\dagger \boldsymbol{b}$ of system (3) may differ much from $\boldsymbol{x}^* = A^\dagger \boldsymbol{b}^*$. Hence special techniques, called *regularization* methods, must be used to obtain acceptable approximations of $\boldsymbol{x}^*$. When iterative methods are employed, they must enjoy the semiconvergence property. According to this property, the initially computed vectors are minimally affected by the noise and approach solution $\boldsymbol{x}^*$. After some iterations, the noise starts to contaminate the computed vectors, which go away from $\boldsymbol{x}^*$. A good terminating procedure is hence needed to stop the iteration. When vector $\boldsymbol{x}_\eta$ has large negative components, general regularization methods may not preserve nonnegativity and special techniques enforcing nonnegativity must be employed.

Many iterative methods used for the deconvolution can be seen as statistical methods, in the sense that they take into account the random nature of the noise. The vector $\boldsymbol{b}$ is described by a model, depending on parameters which characterize the probability distribution of the components of $\boldsymbol{b}$. Maximum likelihood methods estimate the values of the parameters which provide the best fit.

In the case of a white Gaussian noise $\boldsymbol{\eta}$ with zero mean value, the likelihood function to be maximized is

$$\ell(\boldsymbol{x}) = -\frac{1}{2}\,\|\boldsymbol{b} - A\boldsymbol{x}\|^2, \tag{5}$$

where $\|\ .\ \|$ is the Euclidean norm, and in the case of a Poisson noise $\boldsymbol{\eta}$ the likelihood function is

$$\ell(\boldsymbol{x}) = \sum_i \left(b_i \,+\, b_i \log \frac{(A\boldsymbol{x})_i}{b_i} \,-\, (A\boldsymbol{x})_i \right) \tag{6}$$

(see [7] and [4], Ch.7). Letting $f(\boldsymbol{x}) = -\ell(\boldsymbol{x})$, we have then to find the solution $\boldsymbol{x}$ of the constrained problem

$$\begin{cases} \min\ f(\boldsymbol{x}), \\ \boldsymbol{x} \geq \boldsymbol{0}. \end{cases} \tag{7}$$

where the objective function $f(\boldsymbol{x})$ comes from either (5) or (6). Denoting by $\mathrm{grad}(\boldsymbol{x})$ the gradient of $f(\boldsymbol{x})$ and by $H(\boldsymbol{x})$ its Hessian matrix, for the case of the Gaussian noise we have

$$\mathrm{grad}(\boldsymbol{x}) = A^T(A\boldsymbol{x} - \boldsymbol{b}) \quad \text{and} \quad H(\boldsymbol{x}) = A^T A, \tag{8}$$

and for the case of the Poisson noise we have

$$\text{grad}(\boldsymbol{x}) = A^T(\boldsymbol{e} - Y^{-1}\,\boldsymbol{b}) \quad \text{and} \quad H(\boldsymbol{x}) = A^T B Y^{-2} A, \tag{9}$$

where $Y = \text{diag}(A\boldsymbol{x})$ and $B = \text{diag}(\boldsymbol{b})$.

In some contexts the problem (5) is generalized by means of weighted Euclidean norm $\| \, . \, \|_W$, the weight W being a positive definite matrix with a possible statistical meaning [2]. In this case formulas (8) are replaced by

$$\text{grad}(\boldsymbol{x}) = A^T W(A\boldsymbol{x} - \boldsymbol{b}) \quad \text{and} \quad H(\boldsymbol{x}) = A^T W A, \tag{10}$$

In the literature various different techniques are described to solve (7). Many of them, which implement iterative methods coupled with suitable strategies embedding nonnegativity, are special cases of the scaled gradient projection method. Its description, given in Section 3 in a very general form, follows closely the one of [7].

3 The scaled gradient projection (SGP)

The scaled gradient projection method is a generalization of the steepest descent used in unconstrained optimization. The nonnegativity is enforced by projection onto the nonnegative quadrant, i. e. by setting to zero the negative components. A rough sketch of the regularizing algorithm based on this method is given below. Starting from a nonnegative initial point $\boldsymbol{x}^{(0)}$ and relying on a sequence of scaling matrices D_k and of steplengths α_k, it computes a sequence of points $\boldsymbol{x}^{(k)}$. The last $\boldsymbol{x}^{(k)}$ becomes the regularized solution $\boldsymbol{x}_{reg}$.

A Boolean function *stop_cond* specifies when the iteration must be stopped to get a regularized solution (for example by implementing a discrepancy principle). A function *project*, applied to a vector $\boldsymbol{v}$, gives the vector whose ith component is v_i if positive, 0 otherwise. A function *line_search* solves an unconstrained one-dimensional minimization problem.

SGP (scaled gradient projection)

$k = 0$
`repeat`
  select α_k and D_k
  $\boldsymbol{y}^{(k)} = project(\boldsymbol{x}^{(k)} - \alpha_k\, D_k\, \text{grad}(\boldsymbol{x}^{(k)}))$
  $\boldsymbol{p}^{(k)} = \boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}$
  $\lambda_k = line\_search(\boldsymbol{p}^{(k)})$
  $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k\, \boldsymbol{p}^{(k)}$
  $k = k + 1$
`until` *stop_cond*
$\boldsymbol{x}_{reg} = \boldsymbol{x}^{(k)}$

In order to completely define this iterative scheme, the following features must be specified: how to perform the line search, how to choose the scaling matrix D_k and how to update the steplength α_k.

4 The methods

Our aim is to compare the performances of classical descent methods used for solving problem (7) with the performances of some methods [2], [3], [7], [16], [17] recently put to our attention.
– First of all we observe that some classical descent methods (in the following M_1) can be revisited under the framework of SGP (two extensions of classical RNSD are also analyzed).
– Then we consider a class of stationary iterative methods (M_2) obtained by modifying the constrained Cimmino method which implements Jacobi iteration (Cimmino method can be seen as SGP).
– Finally we consider a class of methods (M_3) which implement SGP in the standard form of Section 3, by fully exploiting the different features.

4.1 (M_1) Classical descent methods

All the methods in this class, except EM, are derived from the likelihood function for the Gaussian noise. EM is derived from the Poisson noise. No line search is required since λ_k is always set to 1. For simplicity sake we denote $\boldsymbol{g}^{(k)} = \text{grad}(\boldsymbol{x}^{(k)})$, where $\text{grad}(\boldsymbol{x})$ is the gradient given in (8). Between parentheses are the names used in the figures and tables of Section 6 to identify the different methods.

• *Projected Landweber method (PL)*
It can be seen as an SGP with $D_k = I$ and α_k equal to a constant parameter ω chosen in such a way to assure convergence (in practice $\omega = 1/\sigma_1^2$, where σ_1 is the first singular value of A). Hence

$$\boldsymbol{x}^{(k+1)} = \mathit{project}(\boldsymbol{x}^{(k)} - \omega\, \boldsymbol{g}^{(k)}).$$

• *Projected steepest descent method (PSD)*
It can be seen as an SGP with $D_k = I$ and $\alpha_k = \|\boldsymbol{g}^{(k)}\|^2/\|A\boldsymbol{g}^{(k)}\|^2$. Hence

$$\boldsymbol{x}^{(k+1)} = \mathit{project}\Big(\boldsymbol{x}^{(k)} - \frac{\|\boldsymbol{g}^{(k)}\|^2}{\|A\boldsymbol{g}^{(k)}\|^2}\, \boldsymbol{g}^{(k)}\Big).$$

• *Projected RNSD (PRNSD)*
The residual norm steepest descent method, which is steepest descent applied to system $X_k A^T A\boldsymbol{x} = X_k A^T \boldsymbol{b}$, can be seen as an SGP with $D_k = X_k$ and $\alpha_k = \boldsymbol{g}^{(k)T}\boldsymbol{q}^{(k)}/\|A\boldsymbol{q}^{(k)}\|^2$, where $\boldsymbol{q}^{(k)} = X_k\boldsymbol{g}^{(k)}$. Hence

$$\boldsymbol{x}^{(k+1)} = \mathit{project}\Big(\boldsymbol{x}^{(k)} - \frac{\boldsymbol{g}^{(k)T}\boldsymbol{q}^{(k)}}{\|A\boldsymbol{q}^{(k)}\|^2}\, \boldsymbol{q}^{(k)}\Big).$$

• *ISRA*
The Iterative Space Reconstruction Algorithm can be seen as an SGP with $\alpha_k = 1$ and D_k the diagonal matrix whose ith diagonal entry is $x_i^{(k)}/(A^T A\boldsymbol{x}^{(k)})_i$. In practice, the method is applied as the fixed point recursion

$$x_i^{(k+1)} = x_i^{(k)} \frac{(A^T\boldsymbol{b})_i}{(A^T A\boldsymbol{x}^{(k)})_i}, \quad i = 1, 2, \ldots, N. \tag{11}$$

No projection is required if $\boldsymbol{x}^{(0)} > \mathbf{0}$, since all the components computed by (11) result to be positive.

• *EM*
The Expectation Maximization algorithm can be seen as an SGP, where $\mathrm{grad}(\boldsymbol{x})$ is as in (9), $\alpha_k = 1$ and D_k is the diagonal matrix whose ith diagonal entry is $x_i^{(k)}/(A^T\boldsymbol{e})_i$. In practice, the method is applied as the fixed point recursion

$$x_i^{(k+1)} = \frac{x_i^{(k)}}{(A^T\boldsymbol{e})_i} (A^T\boldsymbol{v})_i, \quad i = 1, 2, \ldots, N, \tag{12}$$

where $\boldsymbol{v}$ is the vector whose ith component is $b_i/(A\boldsymbol{x}^{(k)})_i$. As in the previous case, no projection is required if $\boldsymbol{x}^{(0)} > \mathbf{0}$.

• *MRNSD*
In [15] a modification is suggested to RNSD, which reduces the steplength α_k in such a way to produce only nonnegative components. This modified version of RNSD can also be seen as an SGP with $D_k = X_k$. Letting

$$\boldsymbol{q}^{(k)} = X_k\boldsymbol{g}^{(k)} \quad \text{and} \quad \alpha = \frac{\boldsymbol{g}^{(k)T}\boldsymbol{q}^{(k)}}{\|A\boldsymbol{q}^{(k)}\|^2}, \tag{13}$$

we set

$$\alpha_k = \alpha \quad \text{if} \quad \boldsymbol{g}^{(k)} \leq \mathbf{0} \quad \text{and} \quad \alpha_k = \min\Big(\alpha\,, \frac{1}{\max_i\ g_i^{(k)}}\Big) \quad \text{otherwise.} \tag{14}$$

With this position no projection is required and

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k\ \boldsymbol{q}^{(k)}. \tag{15}$$

• *WMRNSD*
This weighted version of MRNSD has been proposed in [2] and is based on statistical considerations. The method consists in applying the MRNSD algorithm to the weighted version of (5) with

$$W = \mathrm{diag}\Big(\frac{1}{b_1 + \sigma^2},\ \ldots,\ \frac{1}{b_N + \sigma^2}\Big),$$

where σ^2 is the variance of the Gaussian component of the noise. Formulas (13), (14) and (15) still hold, with $\boldsymbol{g}^{(k)}$ replaced by $\widehat{\boldsymbol{g}}^{(k)}$ according to (10), i.e.

$$\widehat{\boldsymbol{g}}^{(k)} = A^T W(A\boldsymbol{x}^{(k)} - \boldsymbol{b}),$$

and the norm in (13) replaced by the weighted norm.

An analysis of the iterates computed by MRNSD and WMRNSD reveals an intrinsic weakness of these methods: if at the kth iteration a component becomes zero, it remains zero for all the subsequent iterations, regardless the correctness of this position. The same thing does not happen with the other methods.

4.2 ($\mathbf{M}_2$) Stationary iterative methods

In the case of Gaussian noise also stationary iterative methods can be used to solve (7). We consider here the following four methods whose convergence properties are proved in the quoted papers.

• *Constrained Cimmino method (CC)*
Jacobi method applied to A^TA with an acceleration parameter ω is known as Cimmino method [16]. It can be seen as an SGP, where $\alpha_k = \omega$, the gradient is as in (10) with

$$W = \mathrm{diag}\Big(\frac{1}{\|A^T\boldsymbol{e}_1\|^2},\ \ldots,\ \frac{1}{\|A^T\boldsymbol{e}_N\|^2}\Big),$$

and $D_k = I$. In practice, the method is applied in the following way

$$\begin{aligned}
&\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{x}^{(k)}\\
&v_i = \frac{r_i}{\|A^T\boldsymbol{e}_i\|^2}, \quad \text{for} \quad i = 1, \ldots, N,\\
&\boldsymbol{x}^{(k+1)} = \mathit{project}\Big(\boldsymbol{x}^{(k)} + \omega\, A^T\boldsymbol{v}\Big)
\end{aligned}$$

• *Constrained Kaczmarz method (CK)*
Gauss-Seidel method applied to AA^T is known as Kaczmarz method [21]. The introduction of the accelerating parameter, as in SOR, is proposed in [18]. For the extension to constrained problems see [17]. It can not be seen as an SGP and is applied in the following way

$$\begin{aligned}
&\boldsymbol{q} = \boldsymbol{x}^{(k)}\\
&\boldsymbol{q} = \boldsymbol{q} + \omega\,\frac{b_i - \boldsymbol{q}^T(A^T\boldsymbol{e}_i)}{\|A^T\boldsymbol{e}_i\|^2}\,A^T\boldsymbol{e}_i, \quad \text{for} \quad i = 1, \ldots, N,\\
&\boldsymbol{x}^{(k+1)} = \mathit{project}(\boldsymbol{q})
\end{aligned}$$

It is clear that for Kaczmarz method the reordering of the rows affects the result. No general recipe is available for finding the best reordering. Since

we want to compare the methods for all the considered problems on a fair common basis, we will adopt the natural increasing ordering of the rows.

Both Cimmino and Kaczmarz methods have been first proposed for consistent problems. Recently extensions for inconsistent problems have been suggested in [16], [17], aiming at reducing the distance between the limiting vector of the iteration and the set of the least squares solutions of (3). Besides the acceleration parameter ω used for updating $\boldsymbol{x}^{(k)}$, another parameter ω' is used in the extended versions.

• *Extended constrained Cimmino method (ECC)*
It can not be seen as an SGP and is applied in the following way

$$
\begin{aligned}
& w_i = \frac{(A^T \boldsymbol{y})_i}{\|A\boldsymbol{e}_i\|^2}, \quad \text{for} \quad i = 1, \ldots, N, \\
& \boldsymbol{y} = \boldsymbol{y} - \omega' A\boldsymbol{w}, \\
& \boldsymbol{r} = \boldsymbol{b} - \boldsymbol{y} - A\boldsymbol{x}^{(k)} \\
& v_i = \frac{r_i}{\|A^T\boldsymbol{e}_i\|^2}, \quad \text{for} \quad i = 1, \ldots, N, \\
& \boldsymbol{x}^{(k+1)} = \mathit{project}\Big(\boldsymbol{x}^{(k)} + \omega\, A^T\boldsymbol{v}\Big),
\end{aligned}
$$

where $\boldsymbol{y} = \boldsymbol{b}$ initially.

• *Extended constrained Kaczmarz method (ECK)*
It can not be seen as an SGP and is applied in the following way

$$
\begin{aligned}
& \boldsymbol{q} = \boldsymbol{x}^{(k)} \\
& \boldsymbol{y} = \boldsymbol{y} - \omega' \frac{\boldsymbol{y}^T(A\boldsymbol{e}_i)}{\|A\boldsymbol{e}_i\|^2} A\boldsymbol{e}_i, \quad \text{for} \quad i = 1, \ldots, N, \\
& \boldsymbol{c} = \boldsymbol{b} - \boldsymbol{y} \\
& \boldsymbol{q} = \boldsymbol{q} + \omega \frac{c_i - \boldsymbol{q}^T(A^T\boldsymbol{e}_i)}{\|A^T\boldsymbol{e}_i\|^2} A^T\boldsymbol{e}_i, \quad \text{for} \quad i = 1, \ldots, N, \\
& \boldsymbol{x}^{(k+1)} = \mathit{project}(\boldsymbol{q}),
\end{aligned}
$$

where $\boldsymbol{y} = \boldsymbol{b}$ initially.

In the general case, finding acceptable estimates of both the optimal values for ω and ω' would be difficult even if the noise does not affect the right hand-side vector. In our case, where the noise is present, both the convergence rate and the reconstruction efficiency depend on the chosen values of the parameters, which are not easily tuned. In addition to the structure of the matrix, also the direction and the size of the noise should be taken into consideration. For Kaczmarz methods in [17] both ω and ω' are set to 1. A preliminary experimentation specifically aimed at determining reasonable values for the parameters has shown that smaller values of ω and ω' give better reconstructions but decrease the convergence rate.

For Cimmino methods in [16] different ω_i and ω'_i are suggested in the theory for the different components i, but in the experiments the values $\omega = \omega' = 2/N$ are always used. For these methods the experimentation has shown that when the values of the parameters increase, the reconstruction first improves then worsens, and that intermediate values are better also from the point of view of the convergence rate. In the introduction of [16] Cimmino methods are presented as especially suited to parallel computers, while Katzmarz methods generally converge faster.

4.3 (M_3) Methods implementing SGP

As we have seen, SGP can be considered as a framework where many descent methods can be formulated. Any scaling matrix scheme and any steplength scheme can be included, as long as the convergence is guaranteed. Convergence proofs can be found in [7], where several combinations of scaling matrices and steplengths are considered for the case of Poisson noise, and in [3] for the case of Gaussian noise. In this section we outline the choices for the line search, the scaling matrix and the steplength we have taken into consideration and tested in the experiments.

4.3.1 The line search

Given a point $\boldsymbol{x}^{(k)}$ and a descent direction $\boldsymbol{p}^{(k)}$, the line search should compute

$$\lambda_k = \arg\min_{\lambda>0} \; f(\mathit{project}(\boldsymbol{x}^{(k)} + \lambda\, \boldsymbol{p}^{(k)})).$$

This problem is typically replaced by the following one

$$\lambda_k = \arg\min_{0<\lambda\leq 1} \; f(\mathit{project}(\boldsymbol{x}^{(k)} + \lambda\, \boldsymbol{p}^{(k)})) = \arg\min_{0<\lambda\leq 1} \; f(\boldsymbol{x}^{(k)} + \lambda\, \boldsymbol{p}^{(k)}), \quad (16)$$

since $\boldsymbol{x}^{(k)} + \lambda\, \boldsymbol{p}^{(k)}$ is nonnegative for $0 < \lambda \leq 1$. When f is quadratic, problem (16) can be exactly solved. In particular, in the case of Gaussian noise we have

$$\lambda_k = \min\Big(1, -\frac{\boldsymbol{p}^{(k)T}\boldsymbol{g}^{(k)}}{\|A\boldsymbol{p}^{(k)}\|^2}\Big). \quad (17)$$

In the case of Poisson noise, problem (16) can be coped with an iterative method (for example Newton's method). Alternatively, an inexact line search can be performed by applying the Armijo rule: an acceptable value λ for λ_k satisfies

$$f(\boldsymbol{x}^{(k)} + \lambda\, \boldsymbol{p}^{(k)}) \leq f(\boldsymbol{x}^{(k)}) + c\, \lambda\, \boldsymbol{p}^{(k)\,T}\mathrm{grad}(\boldsymbol{x}^{(k)}),$$

where $\mathrm{grad}(\boldsymbol{x})$ is the gradient given in (9) and c is a small constant ($c = 10^{-4}$ is usually suggested). By using the expression of $f(\boldsymbol{x})$ given in (6) the Armijo

rule can be rewritten as

$$\sum_i \left[\lambda\,(A\boldsymbol{p}^{(k)})_i - b_i\,\log\left(1+\lambda\,\frac{(A\boldsymbol{p}^{(k)})_i}{(A\boldsymbol{x}^{(k)})_i}\right)\right] \le c\,\lambda\,\boldsymbol{p}^{(k)\,T}\mathrm{grad}(\boldsymbol{x}^{(k)}). \qquad (18)$$

The algorithm starts with $\lambda = 1$, reducing it by bisection until (18) holds.

Variants of the Armijo rule can be found in the literature. For example in [7] a variant is cited, where the value $f(\boldsymbol{x}^{(k)})$ is replaced by the maximum of f on a larger set of preceding points, with the aim of getting a more efficient search. The experiments in [7] do not indicate that this variant is really more efficacious than the original Armijo rule. Several tests we have made confirm this result, together with the fact that getting a more accurate approximation of λ_k by solving accurately (16) does not pay the effort. Hence in our experiments we apply the Armijo rule in the form (18).

4.3.2 The scaling matrix

Concerning the scaling matrix D_k, the following approaches are considered. The first approach, the most trivial one, sets

$$(\mathrm{a}) \qquad \mathrm{D_k} = \mathrm{I}.$$

Next approaches see D_k as a preconditioner for the system $\mathrm{grad}(\boldsymbol{x}) = \boldsymbol{0}$. The introduction of D_k aims at accelerating the convergence rate of the iterative method used to solve problem (7) without a significative increase in the computational cost. This is obtained with a diagonal matrix $D_k = \mathrm{diag}(d_1^{(k)}, \ldots, d_N^{(k)})$. A possible choice is suggested by the Newton-Raphson method and sets D_k equal to the inverse of the diagonal of the Hessian matrix $H(\boldsymbol{x})$. Following this suggestion we have: in the case of the Gaussian noise, from (8)

$$(\mathrm{b}_1) \qquad d_i^{(k)} = \frac{1}{\|A\boldsymbol{e}_i\|^2}, \quad \text{for} \quad i = 1, \ldots, N,$$

where $\boldsymbol{e}_i$ is the ith canonical vector and, in the case of the Poisson noise, from (9)

$$(\mathrm{b}_2) \qquad d_i^{(k)} = \frac{1}{h_i^{(k)}}, \quad \text{where} \quad h_i^{(k)} = \sum_j a_{ji}^2\,\frac{b_j}{(A\boldsymbol{x}^{(k)})_j^2}, \quad \text{for} \quad i = 1, \ldots, N.$$

Setting $\mathrm{grad}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})$, another choice is $d_i^{(k)} = x_i^{(k)}/h_i^{(k)}$, where $h_i^{(k)}$ is the ith component of $\boldsymbol{h}(\boldsymbol{x}^{(k)})$. In the case of the Gaussian noise, letting $\boldsymbol{h}(\boldsymbol{x}) = A^T A\boldsymbol{x}$, we have

$$(\mathrm{c}_1) \qquad d_i^{(k)} = \frac{x_i^{(k)}}{(A^T A\boldsymbol{x}^{(k)})_i}, \quad \text{for} \quad i = 1, \ldots, N,$$

and in the case of the Poisson noise, letting $\boldsymbol{h}(\boldsymbol{x}) = A^T\boldsymbol{e}$, where $\boldsymbol{e}$ is the vector of all ones, we have

$$(c_2) \qquad d_i^{(k)} = \frac{x_i^{(k)}}{(A^T\boldsymbol{e})_i}, \quad \text{for} \quad i = 1, \ldots, N.$$

The scaling matrices (a), (b_1), (b_2) and (c_2) are suggested in [7]. The scaling matrix (c_1) is suggested in [3]. The matrices (c_1) and (c_2) are those used by ISRA and EM respectively.

In any case, D_k must be guaranteed to be positive definite. This is obtained by bracketing the diagonal elements from above and below, in order to get bounded positive entries.

4.3.3 The steplength α_k

The update of steplengths makes use of the two values

$$\alpha_k^{(1)} = \frac{\|D_k^{-1}\boldsymbol{s}^{(k)}\|^2}{\boldsymbol{s}^{(k)T}D_k^{-1}\boldsymbol{z}^{(k)}} \quad \text{and} \quad \alpha_k^{(2)} = \frac{\boldsymbol{s}^{(k)T}D_k\boldsymbol{z}^{(k)}}{\|D_k\boldsymbol{z}^{(k)}\|^2},$$

where $\boldsymbol{s}^{(k)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}$ and $\boldsymbol{z}^{(k)} = \text{grad}(\boldsymbol{x}^{(k)}) - \text{grad}(\boldsymbol{x}^{(k-1)})$. The computed values are forced to be bounded positive.

The following four updating rules are proposed in [7] with the names SGP-BB1, SGP-BB2, SGP-ABB and SGP-SS:

$$(A) \qquad \alpha_k = \alpha_k^{(1)},$$

$$(B) \qquad \alpha_k = \alpha_k^{(2)},$$

$$(C) \qquad \text{if } \alpha_k^{(2)}/\alpha_k^{(1)} \le 0.15 \text{ then } \alpha_k = \alpha_k^{(2)} \text{ else } \alpha_k = \alpha_k^{(1)},$$

$$(D) \qquad \text{if } \alpha_k^{(2)}/\alpha_k^{(1)} \le \tau_k \text{ then } \alpha_k = \min_{j\in[k-2,k]} \alpha_j^{(2)},\ \tau_{k+1} = 0.9\,\tau_k,$$

$$\text{else } \alpha_k = \alpha_k^{(1)},\ \tau_{k+1} = 1.1\,\tau_k,$$

where $\tau_1 = 0.5$.

4.3.4 Combining features

By combining the different features we get two classes of methods: SGP-G based on Gaussian noise and SGP-P based on Poisson noise. In the case of the Gaussian noise the line search is performed applying (17) and the scaling matrices can be (a), (b_1) and (c_1). The corresponding methods are given the names

SGP-Ga, SGP-Gb, SGP-Gc,

further specialized according to the steplength updating. The considered combinations are then

SGP-GaA, SGP-GaB, SGP-GaC, SGP-GaD,
SGP-GbA, SGP-GbB, SGP-GbC, SGP-GbD,
SGP-GcA, SGP-GcB, SGP-GcC, SGP-GcD.

In the case of the Poisson noise the line search is performed by applying Armijo rule which computes (18) at each bisection. The scaling matrices can be (a), (b_2) and (c_2). The corresponding methods are given the names

SGP-Pa, SGP-Pb, SGP-Pc,

further specialized according to the steplength updating. The considered combinations are then

SGP-PaA, SGP-PaB, SGP-PaC, SGP-PaD,
SGP-PbA, SGP-PbB, SGP-PbC, SGP-PbD,
SGP-PcA, SGP-PcB, SGP-PcC, SGP-PcD.

Remarks

1. SGP-Ga methods are generalizations of PL and PSD, SGP-Gc methods are generalizations of ISRA, SGP-Pc methods are generalizations of EM. Usually a generalization aims at improving the convergence rate, but we must remember that in our context a better convergence rate may result in a worst reconstruction efficiency. Actually, this is what happens in some cases, as shown by the experimentation of Section 6.
2. The experimentation of Section 6 will be useful also to verify whether SGP-G methods perform better than SGP-P methods when the right-hand side is contaminated by only Gaussian noise and the other way round when the right-hand side is contaminated by only Poisson noise, and how these methods behave when both noises are present, as in the real cases.

4.3.5 Computational costs

In this section the multiplicative cost of the considered methods is analyzed. We limit the analysis to the cases where the matrix-vector product is computed with a cost $c_A \sim N^2$ (when A has no structure) or $c_A \sim cN \log_2 N$ (for a constant c, when A has a structure which allows the computation of matrix-vector products by means of FFT).

Let c_{it} denote the cost of computing one iteration of a method. For simplicity we assume the cost of computing the logarithm of a floating point number to be roughly equal to the cost of one multiplicative operation.

For M_1 methods, Cimmino methods and SGP-G methods we have

$$c_{it} = \gamma_1 \, c_A + \gamma_2 \, N, \tag{19}$$

and for SGP-P methods we have

$$c_{it} = \gamma_1 \, c_A + (\gamma_2 + 4\mu) \, N, \tag{20}$$

where the constants $\gamma_1 \in [2, 4]$ and $\gamma_2 \in [1, 14]$ (smaller values of γ_2 hold for M_1 and Cimmino methods, larger values for M_3 methods). The constant

μ denotes the number of bisections performed by the Armijo rule, i.e. the number of times formula (18) is computed. The experiments of Section 6 have shown that a reasonable bound for the number $\mu_{\max}$ of allowed bisections is 5, since with larger values the reconstruction efficiency of an SGP-P method is nearly independent from $\mu_{\max}$. In (20) the term in N contains also the cost of computing the logarithm of a vector.

To evaluate the cost of Kaczmarz methods we must take into account that products of single lines of A by a vector (a different vector for each line) are required. Hence, even if A is structured, this computation cannot be performed by means of FFT's. Then, for a matrix represented by a space-invariant PSF with m nonzero elements, we have

$$c_{it} \sim 2\,m\,N \quad \text{for CK} \quad \text{and} \quad c_{it} \sim 4\,m\,N \quad \text{for ECK},$$

while for a matrix with no structure we have

$$c_{it} \sim 2\,N^2 \quad \text{for CK} \quad \text{and} \quad c_{it} \sim 4\,N^2 \quad \text{for ECK}.$$

In the following we assume the size N of A to be large enough to make negligible in (19) and (20) the term in N with respect to the term in c_A. Under this assumption the cost of PL, ISRA, EM, MRNSD, WMRNSD, CC, SGP-G, SGP-Pa and SGP-Pc is $c_{it} \sim 2\,c_A$, the cost of PSD, PRNSD and SGP-Pb is $c_{it} \sim 3\,c_A$ and the cost of ECC is $c_{it} \sim 4\,c_A$.

In the numerical experimentation we will focus on a subset of methods which, in the light of the previous assumption, can be considered roughly equivalent from the point of view of the computational cost per iteration.

5 Performance measures

Numerical simulation is essential to compare the performances of different methods. When performing a simulation, the exact solution $\boldsymbol{x}^*$ is known and for any $\boldsymbol{x}^{(k)}$ computed by a method the relative error

$$\epsilon_k = \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\| / \|\boldsymbol{x}^*\|$$

can be estimated.

We consider several measures (some of them are the discrete version of those defined in [8]). For each problem described in the next section, i.e. a matrix A and a solution $\boldsymbol{x}^*$, many different noisy vectors $\boldsymbol{b}$ are generated from the vector $\boldsymbol{b}^* = A\boldsymbol{x}^*$. Three kinds of noise distributions are considered: only Gaussian noise, only Poisson noise and a combination of both, using different levels of the noise error $\eta = \|\boldsymbol{\eta}\|$ and different proportions of the two noises in the mixed case. A sample for the statistical analysis is given by a pair $(A, \boldsymbol{b})$, for which the noise level η is known. Each method is applied to a sample and the following elements are computed:

– the relative error history ϵ_k and the residual history $\delta_k = \|\boldsymbol{r}^{(k)}\|$, where $\boldsymbol{r}^{(k)} = \boldsymbol{b} - A\boldsymbol{x}^{(k)}$, for $k \geq 0$

– the minimum $\epsilon_{\min}$ of ϵ_k and the corresponding iteration number $k_{\min}$

– the value $\delta_{\min} = \|\boldsymbol{r}^{(k_{\min})}\|$ of the residual norm in $k_{\min}$ and the difference $g_{\min} = \log_{10} \delta_{\min} - \log_{10} \eta$

– the stopping iteration number k_{stop} according to the discrepancy principle, i.e. the smallest index k such that $\|\boldsymbol{r}^{(k)}\| < \eta$, the corresponding stopping error ϵ_{stop} and the ratio $s_{\min} = \epsilon_{\text{stop}}/\epsilon_{\min}$.

Five different measures are obtained by averaging the behaviour of the jth method applied to all the samples of the ith problem. From each set of measures a single indicator is obtained through an averaging procedure.

(a) The *optimal error* $E_{i,j}$, computed by averaging the errors $\epsilon_{\min}$, estimates the reconstruction efficiency of the jth method with respect to the ith problem. The *reconstruction efficiency indicator* E_j of the jth method is obtained by averaging $E_{i,j}$ on i. E_j is expressed in % in the figures and tables of Section 6.

(b) the *optimal iteration number* $K_{i,j}$, computed by averaging the numbers $k_{\min}$. By this quantity we can estimate both the *convergence rate* and the *computational cost* of the jth method, according to values of c_{it} given at the end of Subsection 4.3.5. The *optimal cost indicator* K_j of the jth method is obtained by averaging $K_{i,j}$ on i.

(c) the *stopping iteration number* $F_{i,j}$, computed by averaging the numbers k_{stop}. The *stopping cost indicator* F_j of the jth method is obtained by averaging $F_{i,j}$ on i. This indicator gives a better measure than K_j of the practical cost of the method.

(d) the *consistency measure* $C_{i,j}$, given by the standard deviation of the set of the $g_{\min}$, says if the points $g_{\min}$ lie close to their mean. By this quantity we can estimate if the method is consistent with the discrepancy principle, i.e. if the condition $\delta_{\min} \sim c\ \eta$ holds for a constant c. Graphically, this means that a plot of $\delta_{\min}$ versus η would be nearly rectilinear. The *consistency indicator* C_j of the jth method is obtained by averaging $C_{i,j}$ on i.

(e) the *sensitivity measure* $S_{i,j}$, computed by averaging the ratios $s_{\min}$. By this quantity we can estimate how much the method is sensitive to an incorrect computation of $k_{\min}$ by the discrepancy principle. In practice, low sensitivity means that the error ϵ_{stop} is sufficiently close to $\epsilon_{\min}$. The *sensitivity indicator* S_j of the jth method is obtained by averaging S_{ij} on i.

The effectiveness of the discrepancy principle as a stopping rule requires a good estimate of the noise level to be available, which of course is true in our simulated experiments, but not in the real cases. In our experiments, where the discrepancy condition has been applied with $c = 1$, frequently $F_{i,j} < K_{i,j}$, leading to lower values of F_j with respect to K_j. This indicates that the noise level tends to be overestimated, letting to an underestimation of $k_{\min}$, i.e. to an early stopping of the iteration, which could prevent a good reconstruction. On the contrary, an overestimation of $k_{\min}$ produces in general less damage than an underestimation, because the initial decrease of the error is more pronounced than the subsequent increase after the minimum. In general, large values of S_j indicate that the method is more subjected to underestimate $k_{\min}$. Large values of C_j indicate that the behaviour of the residual norm in the minimum is irregular, leading to a bad estimate of $k_{\min}$. However, a small value of S_j indicates that the reconstruction can be good even if the corresponding C_j is large.

6 Numerical experimentation

The numerical experimentation has been conducted with Java code using double precision arithmetic. Both 1D and 2D test problems have been considered.

6.1 The problems

The 1D problems are obtained from the discretization of Fredholm integral equations of the first kind. They are taken from [11], namely `baart, foxgood, ilaplace, phillips, shaw`. For problem `ilaplace` the 4 examples listed in [11] have been considered. In general, the matrices of these problems are severely ill-conditioned, with more than half the singular values below the machine precision. The size of all the 8 problems is $N = 64$ and for each problem 500 samples have been generated, with relative noise levels from 0.15% to 6.5%. An approximation of σ_1 for Landweber method has been found by a preliminary analysis.

The 2D experimentation deals with images of astronomical and medical interest, widely used in the literature for testing image deconvolution algorithms. For all the images the number of pixels is $N = 128^2$. The PSF's are represented by positive masks normalized in such a way that the sum of the elements is equal to 1. By Grenander and Szegö theorem [10], the largest singular value σ_1 of $A^T A$ is bounded from above by 1. Since all images have sufficiently large zero background along the boundary, the coefficient matrix can be safely approximated by a 2-level circulant matrix.

Four test problems have been considered (see [9] for details on the description of the problems). The first problem deals with an image of the spiral galaxy NGC 1288 blurred by a diffraction-limited PSF. The problem

has been considered in [5], [6]. Noise level varies from 0.4% to 4.2%. The second problem deals with the image of a satellite, which can be found in the package *RestoreTools* [14]. The blur is performed by an exponential mask. Noise level varies from 0.5% to 5%.

The medical images are models of the human brain used in testing the accuracy of the reconstruction algorithms for emission tomography. The third problem deals with a Hoffman phantom [13], which is used for simulating cerebral blood-flow. The blur is performed by a Gaussian PSF. Noise level varies from 0.3% to 3.5%. The fourth problem deals with the Shepp-Logan phantom [22], which contains ellipsis with different absorption properties. The blur is performed by a Gaussian PSF. Noise level varies from 0.6% to 6%.

6.2 Performance evaluation

In order to make the figures and the tables more legible, a selection of the results obtained by running all the considered methods on all the problems is performed on the basis of the pairs (E_j, K_j).

Among the methods of the RNSD family, only WMRNSD is selected, because of its better performance with both 1D and 2D problems.

For M_2 methods good values of the parameters are identified: the values $\omega = \omega' = 1/N$ are chosen for Cimmino methods and $\omega = \omega' = 0.5$ for Kaczmarz methods in the case of 1D problems. They are identified by CC-1, ECC-1,1, CK-0.5 and ECK-0.5,0.5. In the case of 2D problems, M_2 methods are discarded because of their poor performances.

For M_3 methods, the selection chooses the optimal combinations of features. Among SGP-G methods the scaling matrix (b_1) appears to be not worthy for 1D problems. For SGP-P methods the scaling matrix (b_2) is not worthy for both 1D and 2D problems and the scaling matrix (a) is worthy only for 1D problems. In summary, the following methods are selected:
for 1D problems
PL, PSD, ISRA, EM, WMRNSD, CC-1, ECC-1,1, CK-0.5, ECK-0.5,0.5,
SGP-GaD, SGP-GcD, SGP-PaA, SGP-PcA,
for 2D problems
PL, PSD, ISRA, EM, WMRNSD,
SGP-GaB, SGP-GbB, SGP-GcD, SGP-PcB, SGP-PcD.
Actually, other combinations of the steplength updating rules with the scaling matrices have been considered, but their influence on the performance resulted to be very small.

6.3 1D experiments

Figure 1 shows the linear-log plot of the points (E_j, K_j) obtained by the selected methods in the case of mixed noise. Analogous figures for Poisson

only noise or Gaussian only noise are very similar. This answers the question posed in Remark 2 of Section 4.3.4: it seems that there is no advantage in using an SGP-P method when the right-hand side is contaminated only by Poisson noise or in using an SGP-G method when the right-hand side is contaminated only by Gaussian noise. The figure confirms also what

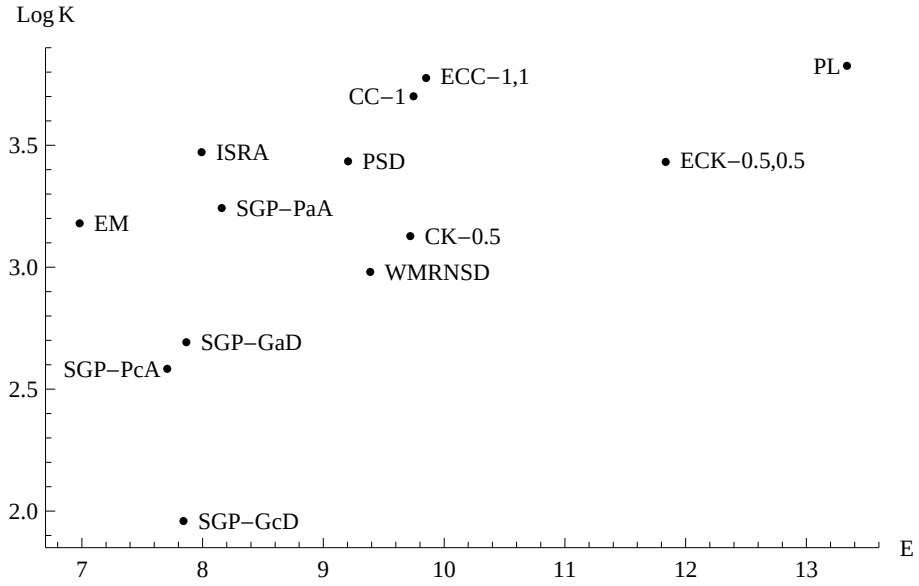

Figure 1: Linear-log (base 10) plot of the points (E_j, K_j) for the 1D problems.

was anticipated in Remark 1 of Subsection 4.3.4: SGP-GaD has a better convergence rate than both PL and PSD, SGP-GcD has a better convergence rate than ISRA, SGP-PcA has a better convergence rate than EM. In the comparison with PL, PSD and ISRA there is also an improvement of the reconstruction efficiency.

As it appears in the figure, the original versions of Cimmino and Kaczmarz methods outperform the extended ones. For this reason the extended versions are discarded in the following table.

Table 1 lists the reconstruction efficiency indicator E_j, the optimal cost indicator K_j, the stopping cost indicator F_j, the consistency indicator C_j and the sensitivity indicator S_j of the jth method.

A first comment is due to explain the poor performance of PL, which is a widely used method for reconstruction problems because of its simplicity. In the experiments a maximum number of 10000 allowed iterations has been set and PL, which has a very low convergence rate, has often been stopped by this bound. This fact has prevented the method from obtaining a proper optimal reconstruction as the values of E_j and C_j point out.

	E_j	K_j	F_j	C_j	S_j
PL	13.34	6698	1621	11.81	1.31
PSD	9.21	2719	734	6.99	1.61
ISRA	7.99	2965	1055	8.09	1.57
EM	6.98	1513	617	7.77	1.53
WMRNSD	9.39	956	651	15.32	1.45
CC-1	9.75	5024	1998	8.05	1.35
CK-0.5	9.72	1340	451	7.88	1.47
SGP-GaD	7.87	492	89	6.29	1.59
SGP-GcD	7.84	91	36	8.22	1.55
SGP-PaA	8.16	1746	682	6.75	1.47
SGP-PcA	7.71	383	136	8.56	1.44

Table 1: Performance indicators of the methods applied to the 1D problems. The lower the indicator, the more performing the method.

A second comment is due to explain the poor consistency measure C_j of WMRNSD. It is produced by an irregular behaviour of the method when the Gaussian noise component is not large enough (the explanation can be found in [2]). As a matter of facts, repeating the experiments with larger Gaussian noise components gives a much smaller value for C_j.

The outstanding position of EM for what concerns the reconstruction efficiency is evident, but EM has a low convergence rate. Hence methods having a worst efficiency but a better convergence rate are of interest, provided that the reconstruction efficiency is not too poor. Taking into account all the performance indicators, some SGP methods, i.e. SGP-GaD, SGP-GcD and SPG-PcA appear to be valid alternatives to EM. In particular, SGP-GcD shows the best convergence rate, with acceptable values of C_j and S_j.

6.4 2D experiments

Figure 2 shows the linear-log plot of the points (E_j, K_j) obtained by the selected methods in the case of mixed noise. As in the 1D case, the experiments confirm that there is no advantage in using an SGP-P method when the right-hand side is contaminated only by Poisson noise or in using an SGP-G method when the right-hand side is contaminated only by Gaussian noise.

The performance indicators are shown in Table 2. The differences among the reconstruction efficiency indicators appear less evident than for the 1D problems, suggesting that the methods, when applied to 2D problems, are roughly equivalent from this point of view. Only a slight better efficiency

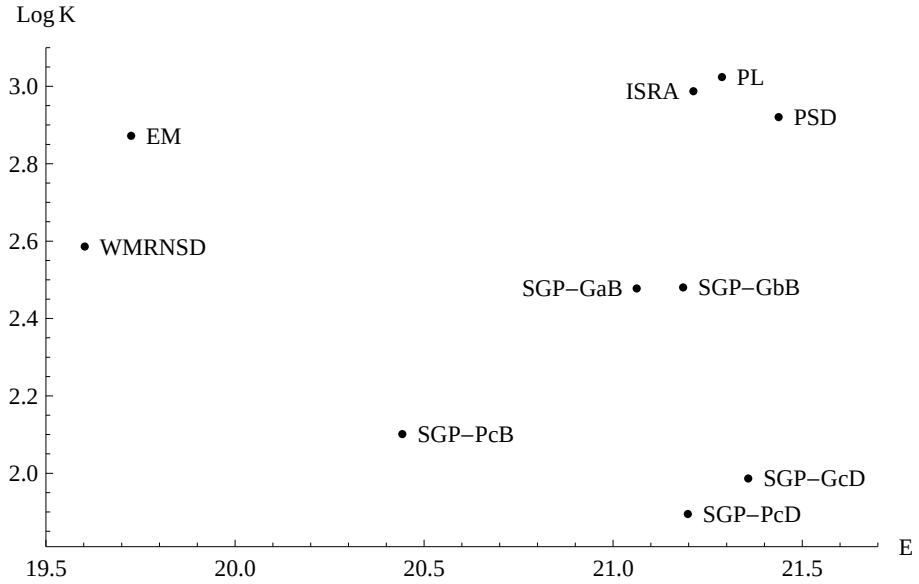

Figure 2: Linear-log (base 10) plot of the points (E_j, K_j) for the 2D problems.

	E_j	K_j	F_j	C_j	S_j
PL	21.29	1057	401	0.96	1.07
PSD	21.44	832	478	0.89	1.06
ISRA	21.21	971	370	1.36	1.06
EM	19.73	745	460	0.85	1.08
WMRNSD	19.60	385	220	1.05	1.07
SGP-GaB	21.06	300	130	0.82	1.07
SGP-GbB	21.19	302	131	0.93	1.08
SGP-GcD	21.36	97	61	1.99	1.05
SGP-PcB	20.44	126	68	1.17	1.08
SGP-PcD	21.20	79	39	3.26	1.08

Table 2: Performance indicators of the methods applied to the 2D problems. The lower the indicator, the more performing the method.

of EM and WMRNSD can be noted. The good performance of WMRNSD does not disagree with its behaviour in the 1D case. In fact, even if the relative noise levels vary in ranges similar to those of the 1D problems, the absolute noise levels are here higher than in the 1D case. Consequently, the Gaussian component of the absolute noise is larger and according to [2] this justifies a better performance of the method.

When we consider the cost indicators, the SGP methods appear to be

preferable. Hence they can be considered valid alternatives to EM, even if they have larger values of C_j, thanks to their small values of S_j. Both SGP-GcD and SGP-PcD show very good convergence rates, while WMRNSD should be preferred if efficiency is the primary concern.

Figure 3 refers to the reconstruction of the Shepp-Logan image corrupted by a relative noise level of 3%. The optimal reconstructions require 41 iterations by SGP-PcD and 327 iterations by EM, and are affected by a relative error of 32.5% and 31.4% respectively. Comparing the two reconstructions, we see that SGP-PcD produces more artifacts than EM. The better reconstruction of EM is paid by many more iterations.

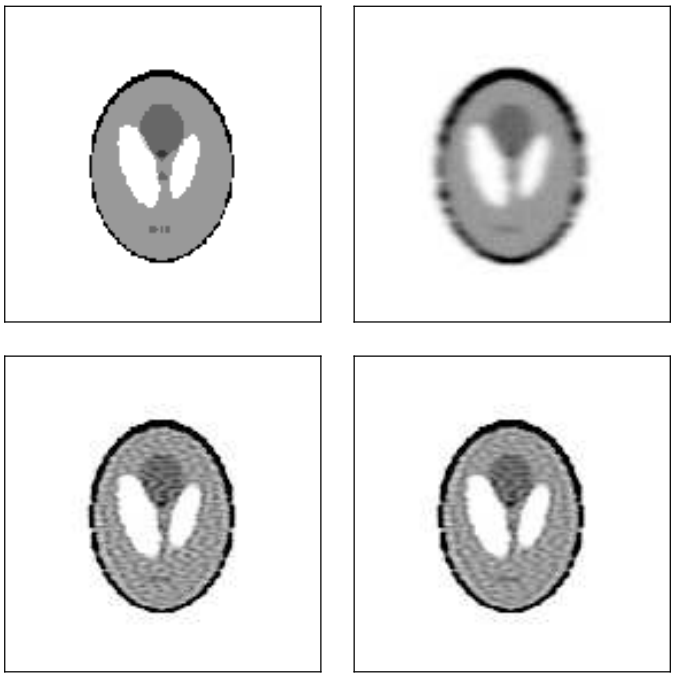

Figure 3: Original Shepp-Logan image (upper left), noisy image with a relative noise level of 3% (upper right), reconstructed image with 41 iterations of SGP-PcD (lower left) and reconstructed image with 327 iterations of EM (lower right).

7 Conclusions

In this paper we have revisited, under a statistical unifying approach, several iterative methods for solving regularization problems with nonnegativity constraints. An extensive experimentation has been carried out in order to compare these methods from different points of view, as the reconstruction efficiency, the computational cost, the consistency with the discrepancy

principle and the sensitivity to the choice of the regularization parameter.

The results of the experimentation for 1D and 2D problems indicate that EM outperforms the other M_1 methods from both efficiency and computational cost points of view, and that M_3 methods with suitable features combinations result to be valid alternatives to EM from the computational cost point of view, maintaining a good efficiency. In particular, the best SGP methods are those which can be seen as generalizations of PL, ISRA and EM. In the 2D case also WMNRSD, which is a generalization of RNSD, can be taken into consideration for its efficiency. These methods lend themselves well to the use of the discrepancy principle as a stopping rule, provided that a reliable estimate of the noise level is available.

References

[1] Archer G E B and Titterington D M 1995 The iterative image space reconstruction algorithm (ISRA) as an alternative to the EM algorithm for solving positive linear inverse problems *Statistica Sinica* **5** 77-96

[2] Bardsley J M and Nagy J G 2006 Covariance-preconditioned iterative methods for nonnegatively constrained astronomical imaging *Siam J. Matrix Anal. Appl.* **27** 1184-1197

[3] Benvenuto F, Zanella R, Zanni L and Bertero M 2010 Nonnegative least-squares image deblurring: improved gradient projection approaches *Inverse Problems* **26** 025004.

[4] Bertero M and Boccacci P 1998 *Introduction to Inverse Problems in Imaging* (Bristol: Institute of Physics Publishing)

[5] Bertero M and Boccacci P 2000 Image restoraton methods for the Large Binocular Telescope (LBT) *Astron. Astrophys. Suppl. Ser.* **147** 323-333

[6] Bertero M, Boccacci P, Correia S and Richichi A 2000 Tomographic methods for the restoration of LBT images *Interferometry in Optical Astronomy* eds Lena P J and Quirrenbach A, Proc SPIE, vol 4006

[7] Bonettini S, Zanella R and Zanni L 2009 A scaled gradient projection method for constrained image deblurring *Inverse Problem* **25** 015002

[8] Brianzi P, Favati P, Menchi O and Romani F 2006 A framework for studying the regularizing properties of Krylov subspace methods *Inverse Problems* **22** 1007-1021

[9] Favati P, Lotti G, Menchi O and Romani F 2006 Iterative Image Restoration with nonnegativity constraints, *Technical Report IIT TR-10/2006*
http://www.iit.cnr.it/attivita/pubblicazioni/pubblicazioni_new.php

[10] Grenander U and Szegö G 1984 *Toeplitz Forms and their Applications*, 2nd edition (N. Y.: Chelsea).

[11] Hansen P C 1994 Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems *Numerical Algorithms* **6** 1-35

[12] Hansen P C 1998 *Rank-Deficient and Discrete Ill-Posed Problems* (Philadelphia: SIAM Monographs on Mathematical Modeling and Computation)

[13] Hoffman E J, Cutler P D, Digby W M, Mazziotta J C 1990 3-D phantom to simulate cerebral blood flow and metabolic images for PET *IEEE Trans. Nucl. Sci.* **37** 616-620

[14] Lee K P, Nagy J G and Perrone L 2002 *Iterative methods or image restoration: a Matlab object oriented approach* http://www.mathcs.emory.edu/∼nagy/RestoreTools

[15] Nagy J and Strakos Z 2000 Enforcing nonnegativity in image reconstruction algorithms *Math. Model., Estim., and Imag.* Wilson D C et al, Eds vol 4121 182-190

[16] Petra S, Popa C and Schnorr C 2008 *Extended and Constrained Cimmino-type Algorithms with Applications in Tomographic Image Reconstruction* IWR Preprint No. 8798, University of Heidelberg. (http://www.uniheidelberg. de/archiv/8798)

[17] Popa C 2008 Constrained Kaczmarz extended algoritm for image reconstruction *Linear Algebra and its Applications* **429** 2247-2267

[18] Popa C and Zdunek R 2004 Kaczmarz Extended Algorithm for Tomographic Image Reconstruction from Limited-Data *Mathematics and Computers in Simulation* **65** 579-598

[19] Pruksch M and Fleischmann F 1998 Positive Iterative Deconvolution in Comparison to Richardson-Lucy Like Algorithms *Astronomical Data Analysis Software and Systems VII ASP Conference Series* Albrecht R, Hook R N and Bushouse H A, Eds vol 145

[20] Saad Y 1996 *Iterative Methods for Sparse Linear Systems*, (Boston: PWS Publishing Co)

[21] Tanabe K 1971 Projection Method for Solving a Singular System of Linear Eqtions and its Applications *Numer. Math.* **17** 203-214

[22] http://www.oersted.dtu.dk/ftp/jaj/31655/ct_programs/